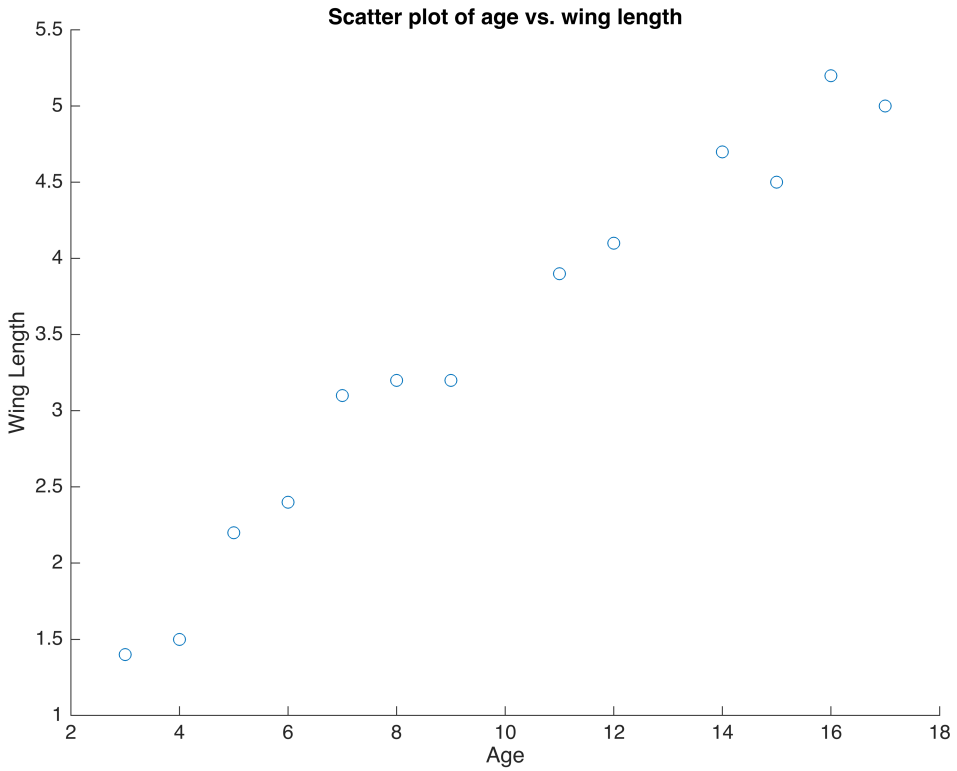
HW: Linear regression

```
% Create a table containing the data
age = [3, 4, 5, 6, 7, 8, 9, 11, 12, 14, 15, 16, 17];
wing_length = [1.4, 1.5, 2.2, 2.4, 3.1, 3.2, 3.2, 3.9, 4.1, 4.7, 4.5, 5.2,
5.0];
data = table(age', wing_length', 'VariableNames', {'Age', 'WingLength'});
n = 13
```

```
n = 13
```

```
% Plot the relationship between age and wing length
hold on;
scatter_plot = scatter(age, wing_length);
xlabel('Age');
ylabel('Wing Length');
title('Scatter plot of age vs. wing length');
```

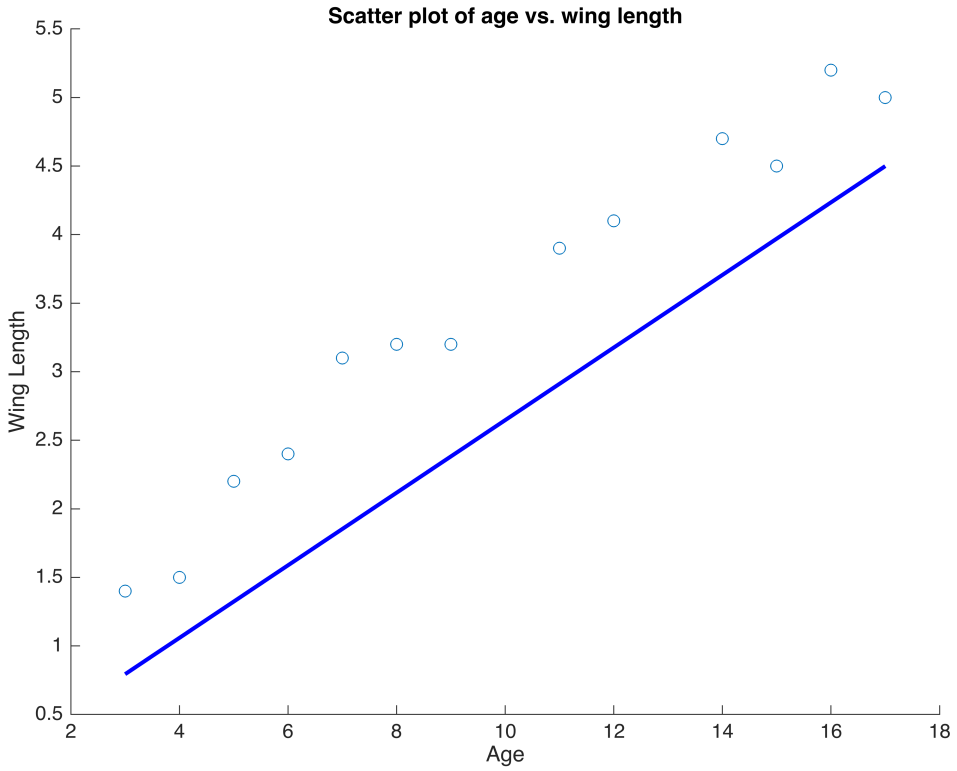



```
% Calculate the regression line using 'polyfit' function

    % Fit a linear regression line (a first-degree polynomial)
    coefficients = polyfit(age, wing_length, 1);

    % The coefficients(1) represents the slope of the linear regression line
    slope = coefficients(1);
```

```
    % Display the slope
    disp(['The slope of the linear regression line is: ', num2str(slope)]);
```

```
The slope of the linear regression line is: 0.26468
```

```
% Plot the regression line

    % Calculate the y-values for the regression line based on the slope
    % and the x-values from the data
    x_regression = age;  % x-values from data
    y_regression = slope * x_regression;

    % Create a line for the regression
    regression_line = line(x_regression, y_regression, 'LineWidth', 2,
'Color', 'blue');
```




```
% Fit a linear regression model to your data
X = ones(size(age));  % Include a column of ones for the intercept?
lm = fitlm(age, wing_length); % did not use 'X' because it gave a rank
deficient warning
```

```
% Perform the F-test for overall model fit (null hypothesis: all
coefficients are zero)
anova_results = anova(lm);

% Display the p-value
disp(anova_results)
```

	SumSq	DF	MeanSq	F	pValue
x1	18.937	1	18.937	289.4	3.0097e-09
Error	0.71979	11	0.065436		

```
% Calculate the confidence intervals for the regression coefficients
CI = coefCI(lm);

% Extract the lower and upper bounds of the confidence intervals for the
slope
lower_bound_slope = CI(2, 1);  % 2 corresponds to the slope coefficient, 1
for lower bound
upper_bound_slope = CI(2, 2);  % 2 corresponds to the slope coefficient, 2
for upper bound

% Plot the confidence intervals for the regression line
lower_CI = line(x_regression, lower_bound_slope * x_regression,
'LineWidth', 1, 'Color', 'red');
upper_CI = line(x_regression, upper_bound_slope * x_regression,
'LineWidth', 1, 'Color', 'red');

% Legend for the plot
legend('Regression Line', 'Lower CI', 'Upper CI');

hold off;
```


```
% Display the R-squared value as calculated by 'fitlm'
disp(['R-squared: ', num2str(lm.Rsquared.Ordinary)]);
```

```
R-squared: 0.96338
```

```
% Calculate Pearson's correlation coefficient
corr_coefficient = corr(data.Age, data.WingLength);

% Display Pearson's correlation coefficient
disp(['Pearson''s correlation coefficient (r): ',
num2str(corr_coefficient)]);
```

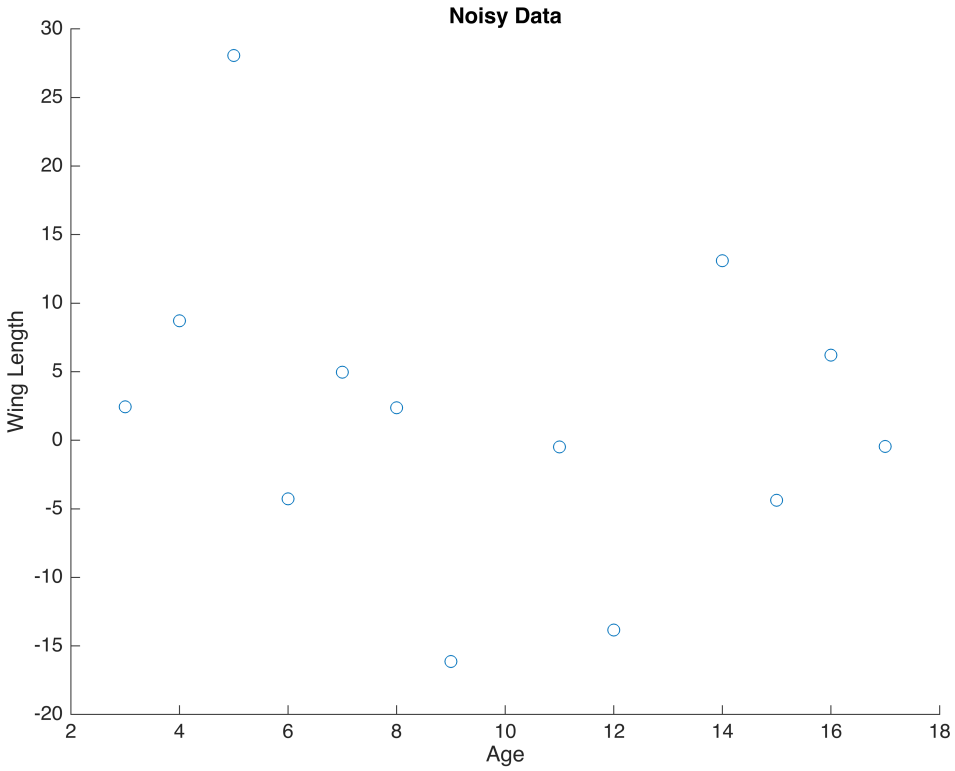
```
Pearson's correlation coefficient (r): 0.98152
```

```
% Add noise to the data
    % Define the standard deviation of the noise
    noise_std = 10;

    % Generate random Gaussian noise
    noise = noise_std * randn(size(wing_length));

    % Add noise to the 'WingLength' data
    noisy_wing_length = wing_length + noise;
```

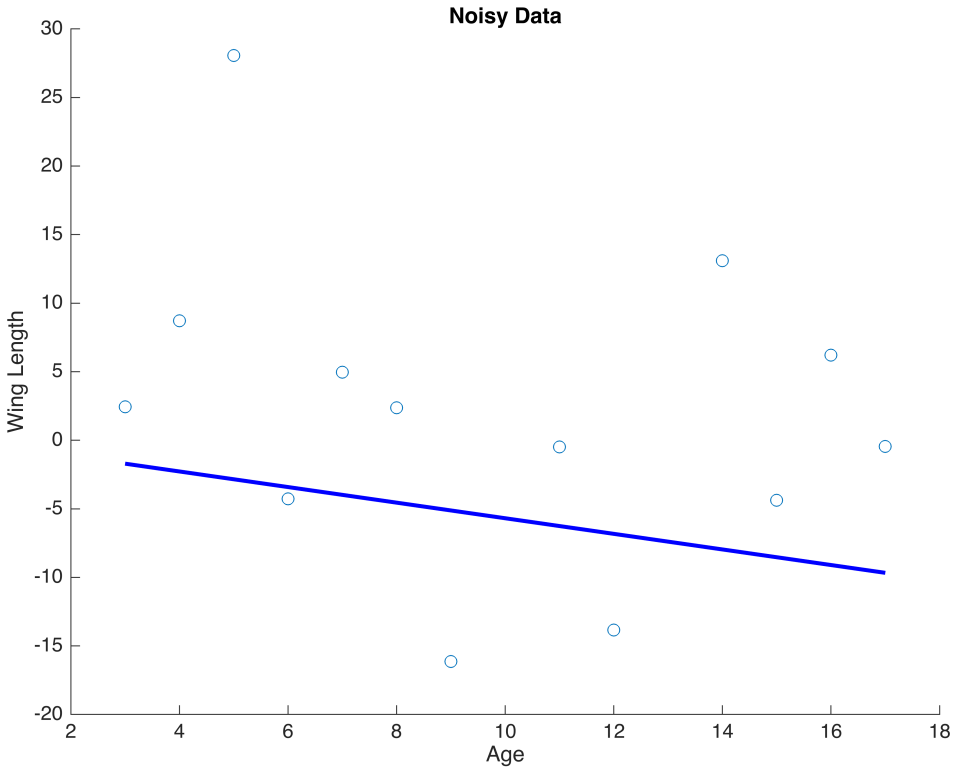
```
    % Create a new table with the noisy data
    noisy_data = table(age', noisy_wing_length', 'VariableNames', {'Age',
'NoisyWingLength'});


% Calculate the regression line using 'polyfit' function
    % Fit a linear regression line (a first-degree polynomial)
    coefficients2 = polyfit(age, noisy_wing_length, 1);

    % The coefficients(1) represents the slope of the linear regression line
    slope2 = coefficients2(1);

    % Display the slope
    disp(['The slope of the linear regression line is: ', num2str(slope2)]);
```

```
The slope of the linear regression line is: -0.5687
```

```
% Plot the data
figure;
    scatter(age, noisy_wing_length);
    hold on;
    xlabel('Age');
    ylabel('Wing Length');
    title('Noisy Data');
```




```
    % Calculate the y-values for the regression line based on the slope
    % and the x-values from the data
    x_regression2 = age;  % x-values from data
    y_regression2 = slope2 * x_regression2;

    % Create a line for the regression
    regression_line2 = line(x_regression2, y_regression2, 'LineWidth', 2,
'Color', 'blue');
    hold off
```


Noisy Data
Wing Length
Age
30
25
20
15
10
5
0
-5
-10
-15
-20
2
4
6
8
10
12
14
16
18